CLAIMS

What is claimed is:

1. A method for revising a software application wherein said software application utilizes persistent data, said method comprising:

applying an upgrade to a first next level of software that understands both old and new persistent data structure formats;

converting all persistent data structures into the old persistent data structure format;

applying an upgrade to a second next level of software that understands said old and new persistent data structure formats; and

converting all persistent data structures into the new persistent data structure format.

2. The method of claim 1, wherein said persistent data structures comprise communication packet structures.

3. The method of claim 2, wherein said software application comprises a distributed system software application including a plurality of nodes holding non-volatile memory data structures.

4. The method of claim 3, wherein said nodes communicate with one another.

5. The method of claim 4, wherein the communication between said nodes occurs using said communication packet structures.

6. The method of claim 1, further comprising:

applying a downgrade to a first previous level of software that understands both said old and new persistent data structure formats;

converting all persistent data structures into the old persistent data structure format; and

applying a downgrade to a second previous level of software that understands said old persistent data structure formats.

7. A system for providing updates to a software application wherein said software application utilizes persistent data, said system comprising:

a first module operable for applying an upgrade to a first next level of software that understands both old and new persistent data structure formats;

a first converter in said first module operable for converting all persistent data structures into the old persistent data structure format;

a second module operable for applying an upgrade to a second next level of software that understands said old and new persistent data structure formats; and

a second converter in said second module operable for converting all persistent data structures into the new persistent data structure format.

8. The system of claim 7, wherein said persistent data structures comprise communication packet structures.

9. The system of claim 8, wherein said software application comprises a distributed system software application including a plurality of nodes holding non-volatile memory data structures.

10. The system of claim 9, wherein said nodes communicate with one another.

11. The system of claim 10, wherein the communication between said nodes occurs using said communication packet structures.

12. The system of claim 7, further comprising:

a third module operable for applying a downgrade to a first previous level of software that understands both said old and new persistent data structure formats;

a third converter in said third module operable for converting all persistent data structures into the old persistent data structure format; and

a fourth module operable for applying a downgrade to a second previous level of software that understands said old persistent data structure formats.

13. A system for providing updates to a software application wherein said software application utilizes persistent data, said system comprising:

means for applying an upgrade to a first next level of software that understands both old

and new persistent data structure formats;

means for converting all persistent data structures into the old persistent data structure format;

means for applying an upgrade to a second next level of software that understands said old and new persistent data structure formats; and

means for converting all persistent data structures into the new persistent data structure format.

14. The system of claim 13, further comprising:

means for applying a downgrade to a first previous level of software that understands both said old and new persistent data structure formats;

means for converting all persistent data structures into the old persistent data structure format; and

means for applying a downgrade to a second previous level of software that understands said old persistent data structure formats.

15. A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method for revising a software application wherein said software application utilizes persistent data, said method comprising:

applying an upgrade to a first next level of software that understands both old and new persistent data structure formats;

converting all persistent data structures into the old persistent data structure format;

applying an upgrade to a second next level of software that understands said old and new persistent data structure formats; and

converting all persistent data structures into the new persistent data structure format.

16. The program storage device of claim 15, wherein said persistent data structures comprise communication packet structures.

17. The program storage device of claim 16, wherein said software application comprises a distributed system software application including a plurality of nodes holding non-volatile memory data structures.

18. The program storage device of claim 17, wherein said nodes communicate with one another.

19. The program storage device of claim 18, wherein the communication between said nodes occurs using said communication packet structures.

20. The program storage device of claim 15, wherein said method further comprises:

applying a downgrade to a first previous level of software that understands both said old and new persistent data structure formats;

converting all persistent data structures into the old persistent data structure format; and

applying a downgrade to a second previous level of software that understands said old

6 persistent data structure formats.